

Article

A Layered Architecture for Concurrent CSI-Based Applications in Smart Environments

Shervin Mehryar 

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, ON M5S 3G4, Canada; shervin.mehryar@utoronto.ca

Abstract

The prevalence of radio frequency signals in indoor environments has in recent years given rise to new technologies across many domains such as robotics, healthcare, and surveillance. Radio frequency signals propagate in the wireless medium through multiple paths and carry useful environment-dependent information. Capturing and analyzing these signal patterns can offer new solutions for a number of applications relevant to ranging, tracking, perception and recognition. In this work we propose a novel architecture, separating physical, back-bone networks, and inference layers, towards fully ubiquitous passive recognition systems that scale with the number of environments and applications. We propose a back-bone architecture that utilizes a novel Cross Dual-Path Attention (CDPA) block to capture spatial and temporal correlations from Channel State Information (CSI) for device-free, multi-task applications. Subsequently, a distill and transfer algorithm is proposed to generalize the inference capabilities of CDPA over multiple target environments for scalable training and reduced computational costs. By sharing knowledge between models across a shared network, experimentation shows that edge devices can be deployed with improved performance while simultaneously meeting strict computation and memory requirements. Our distributed learning paradigm demonstrates that CDPA-based models are capable of using passive signals in a non-intrusive and privacy-protecting manner, in order to achieve ubiquitous recognition at scale in smart environments.

Keywords: channel state information; passive sensing; cross-dual path attention; distillation; knowledge transfer; smart applications; activity; pose; gesture; direction; location; IoT



Academic Editors: Chunjong Zhang,
Weiwei Jiang and Tao Xie

Received: 4 July 2025

Revised: 26 August 2025

Accepted: 13 October 2025

Published: 17 February 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The prevalence of radio frequency signals has in recent years given rise to new solutions in emerging indoor IoT applications across many domains. In indoor environments, radio frequency signals propagate in the wireless medium through multiple paths, bouncing off different objects, and carrying useful environment-dependent information [1]. By capturing and analyzing the signal patterns, subject activities for instance are now detectable in a process termed passive sensing [2]. With the continuing increase in number and computing power of IoT edge devices, the possibility to leverage passive sensing opportunistically is on the rise. A number of other applications including pose estimation [3], gesture recognition [4], and activity recognition and monitoring [5] are now making use of this emerging technology. As such, passive sensing can play an integral role in future context-aware computing across many domains, including health care, robotics, and surveillance [6].

Compared to other WiFi-based recognition modalities, including device-based and camera-based approaches, device free (or passive) recognition offers a number of key advantages. In device-based approaches for example, input radio signals are collected directly from sensors attached onto subjects. Equipping subjects with digital sensors is inherently cumbersome, and in residential and most commercial applications impractical. Camera-based solutions, although circumventing this hassle, bear the risk of privacy infringement, making them less attractive over non-visual solutions. Alternatively, passive sensing from radio frequency signals transceived between edge-devices in indoor settings is remarkably non-intrusive (i.e., no wearables), privacy protective (i.e., no cameras), and of high enough resolution (due to multi-antenna and frequency division technology).

Considering the advantages of device-free solutions, recently the channel state information (CSI) as a source of input has gained much attention [7]. For passive sensing with CSI, all that is required is a wireless channel (for example at 2.4 GHz or 5 GHz in the 802.11ax standard) established between various WiFi enabled devices which readily exist in many indoor environments (e.g., laptops, tablets, etc.) at no additional cost. Because the movement of human subjects in relation to objects changes the multipath characteristics of the channel, the CSI will have a different amplitude/phase at any given time. As opposed to received signal strength (RSS) which provides coarse information (MAC layer), the CSI carries fine-grained information (PHY layer) measured through orthogonal frequency division multiplexing (OFDM) on multiple sub-carriers [8], which makes it ideal as a multi-dimensional source of input. Other sources such as ultra-wide band and continuous wave radar also exist often at an additional per device cost [9]. The CSI nevertheless contains sufficient features, is accessible, economical, and therefore highly lucrative for passive edge computing.

Building fully ubiquitous passive recognition systems from a technological standpoint however remains a challenge due to the fact that small changes in environment settings greatly impact the received patterns. The predominant factors here are: (1) equipment setting (e.g., distance and antennas/sub-carrier specifications); (2) environment artifacts (e.g., furniture layout); (3) location of devices and subjects; (4) pose/orientation of subjects with respect to transceivers. These effects can vary within a given environment, for instance when furniture are rearranged, as well as between environments, for instance in a different room. As a result, a recognition model that is trained in one specific environment may not work well in another environment. Device-based approaches can easily treat such effects as background noise by the virtue of capturing correlation signals directly through sensors. In passive sensing however, there is no straight forward way for separating the environment effects from the subject's movement patterns, a criterion necessary for any practical indoor applicability.

In order to learn domain-specific characteristics while simultaneously sharing their common features, deep learning methods can be leveraged for multi-environment scenarios in a distributed fashion [9]. In this area distributed deep learning research has shown promising performance based on rich pattern extraction in the CSI data, as these methods offer data-driven and automatic feature learning over traditional approaches (i.e., feature engineering). However most reported results are in controlled research environments with training data re-collected to handle different cases. In most practical settings, emulating and labeling all possible signal variations are unrealistic [10]. Another limitation in ubiquitous computing arises due to model diversity, in which different edge devices may be required to run with a different architecture. Therefore traditional distributed techniques such as federated learning may not directly be applicable despite their high efficiency [11,12].

To address these challenges, in this work we separate the task of application-dependent classification from the task of adaptability to new settings in a layered approach. In one

layer, we design an inference and classification architecture with multiple blocks focusing on CSI feature processing and extraction covering a wide range of recognition tasks. In the next layer, we focus on domain (defined by a pair of environment and human subject) adaptation. To this end, we introduce a hybrid federated and mutual distillation training algorithm that is able to bring performance up to the level of a fully centralized model. Our contributions are as follows:

- a layered architecture to, locally and globally, achieve consistent accuracies using edge computation and support multiple applications.
- a transfer policy to ensure commonly learned features are passed to new environments for efficient training through diversified architectures.

In what follows we provide an overview of the proposed framework as shown in Figure 1. The framework consists of four layers: (1) application layer to support multiple indoor tasks; (2) the generalization layer to distribute learned features (domain independent); (3) the backbone network layer for domain adaptation to specific tasks; (4) physical layer which consists of the edge devices and network capture cards. Through different layers, the proposed design offers performance gains by isolating preprocessing, feature extraction, and inference, achieving higher accuracy than non-layered state-of-the-art models across tasks. It provides strong adaptability, as the generalization layer can seamlessly incorporate new tasks by reusing backbone features, enabling concurrent multi-task learning across diverse datasets [12]. The framework also ensures scalability, with lightweight models that can be deployed and fine-tuned efficiently in new environments at low training cost. Lastly, it is able to improve efficiency through feature reuse, reducing computational overhead, while leveraging teacher–student distillation and transfer learning to optimize deployment under resource constraints. In the remaining subsections, each component is further elaborated on.

1.1. Physical Layer

IEEE 802.11n/ac wireless devices leverage multi-input multi-output technology with several transmit and receive antennas to enhance diversity, array, and multiplexing gains, while mitigating co-channel interference. They employ OFDM modulation, dividing bandwidth into orthogonal subcarriers, each experiencing flat fading to counter small-scale fading effects. CSI is estimated by dividing the output signal by a known pilot sequence, offering subcarrier-specific channel dynamics. CSI amplitude is reliable for feature extraction and classification, remaining stable but sensitive to transmission power and rate, with burst noise that is reducible through filtering. CSI phase, on the other hand, is impacted by carrier frequency offset and sampling frequency offset, varying by subcarrier, which obscures small phase changes [13].

In particular for multi-task classification in a given setting, the CSI amplitude profile of a communication channel includes the effect of scattering, fading, and distance-related power attenuation caused by movement. These inherent spatial and temporal dependencies related to location, pose, and motion, can be leveraged to provide high quality features using deep learning. Although filtering techniques exist for phase evaluations for finer motions [14,15], this paper directly focuses on raw CSI amplitudes for richer feature learning and leaves advanced phase correction as future research. In order to learn useful representation for multi-task classification, we then introduce modules with which temporal and spatial features from amplitude data across different tasks can be learned, including activity, pose, gestures and micro-motions, as explained in the next section.

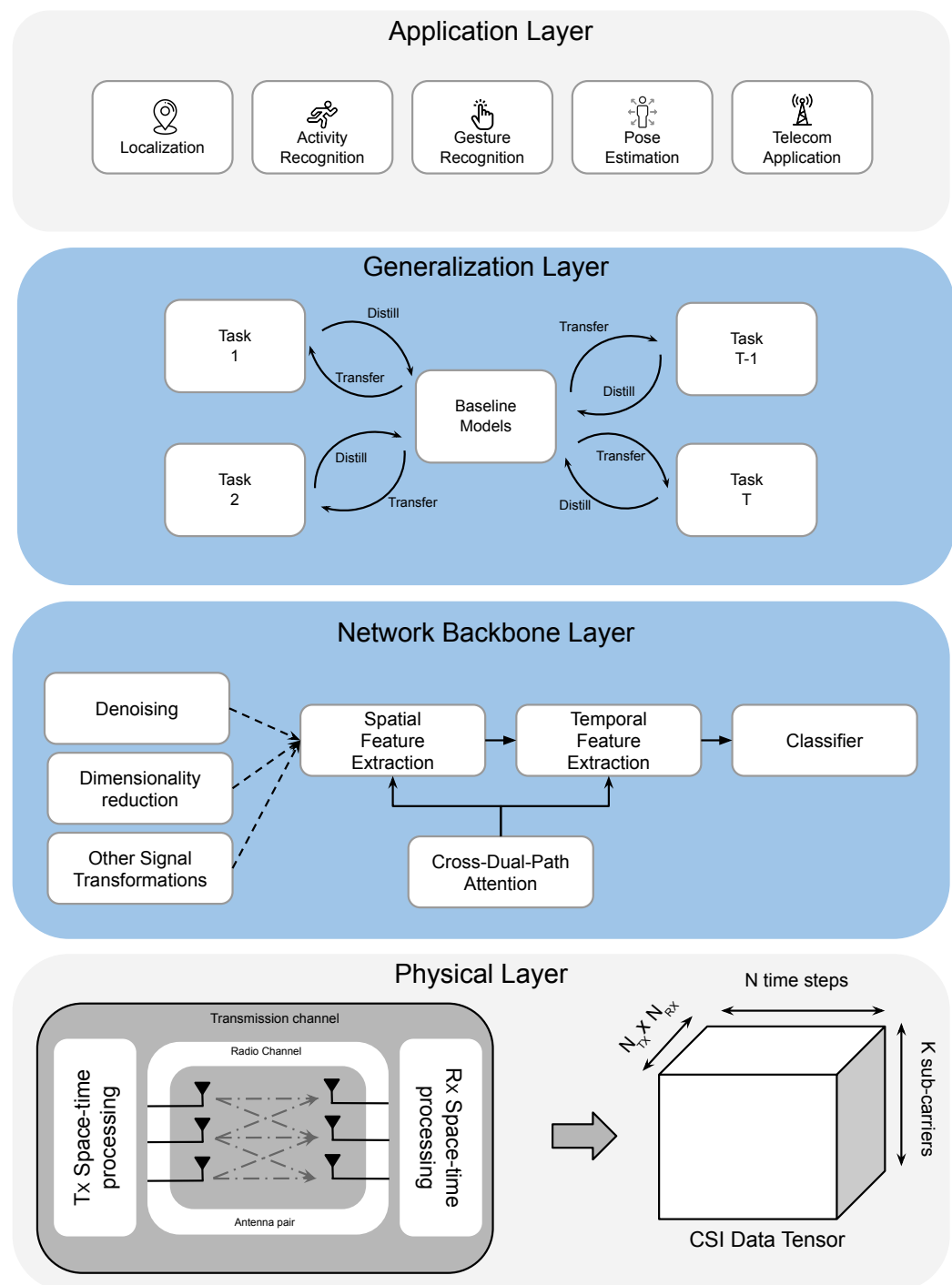


Figure 1. A layered framework for CSI-based passive sensing and concurrent applications.

1.2. Network Backbone Layer

Researchers in recent years have been a strong proponent for the use of machine learning opportunistically for recognition tasks through passive CSI packet collection [16]. A multitude of methods have taken advantage of this approach and shown promising results predominantly for activity monitoring (e.g., LSTM [2], Bi-LSTM [16], ViT [17], and GAN [18]). In multi-room scenarios where activity can vary in different rooms, hybrid Convolutional Neural Networks (CNN) plus Long-Short-Term-Memory (LSTM) classifiers [19] as well as attention networks [16], have been proposed to track variations in action. To diversify the categorical output of the networks to include small gestures (e.g., swiping versus standing, sitting versus sit-down), device based algorithms exist that require wear-

able sensors for the task [20–22]. In device-free case, according to the work in [19], Fourier representations can be incorporated to achieve a similar outcome. In [23], a central model with multi-modal frequency fusion is developed to learn from coarse and fine features and in [24] device-free pose estimation is proposed. Other important factors for indoor applications are related to device locations as well as background environment settings (e.g., furniture). Work has been ongoing for the purpose of localization and fingerprinting with CSI [25–27]. A summary of related models categorized by different tasks is provided in Table 1.

Table 1. Summary of baseline models for CSI-Based tasks—AR: Activity Recognition, GR: Gesture Recognition, PR: Pose Recognition, LFP: Location Finger-Printing.

Author	Year	Abbreviation	Methodology	Task
Saidahmed et al. [28]	2019	ABLSTM	Bi-directional Long-Short Term Memory	GR
Shalaby et al. [29]	2022	CNN-GRU	CNN+GRU for spatiotemporal feature extraction	AR
Mehryar et al. [10]	2022	LA-DCML	Dual modality for proximity and activity	AR, LFP
Islam et al. [30]	2023	STC-NLSTMNet	CNN+LSTM for spatiotemporal feature extraction	AR
Jannat et al. [31]	2023	AAE+RF	Statistical feature extraction with amplitude and phase	AR
Huang et al. [32]	2023	DiffAR	Adaptive Conditional Diffusion Model	GR
Zhang et al. [33]	2023	ImgFi	Gramian Angular Sum and Difference Fields	GR
Yang et al. [34]	2023	WiTransformer	Spatiotemporal Transformer architecture	GR
Bhat et al. [35]	2023	MM-Pose	Pose skeleton from mmWave CSI	PR
Chen et al. [23]	2024	MSF-Net	Dual-stream structure with STFT and DWT	GR, AR
Bhat et al. [36]	2024	GAN-mmWave	GAN-Augmented mmWave CSI	PR
Qin et al. [37]	2024	WinDR	Antenna feature mapping model	PR, GR
Quy et al. [38]	2025	PA-CSI	Spatiotemporal Gated Residual Networks	AR
Wakili et al. [39]	2025	BiLSTM	Bi-directional Long-Short Term Memory	GR
Hu et al. [40]	2025	RGANet	Residual networks (ResNet) with GRU	GR

Despite these advances, building fully ubiquitous passive recognition systems that can support multiple applications remains challenging in practice due to physical-layer characteristics, edge device constraints, and environment-specific variability. The backbone layer is designed to address this challenge by enabling concurrent feature learning across a wide variety of tasks—including direction, pose, and location estimation—that differ significantly in preprocessing requirements (e.g., denoising, dimensionality reduction, signal transformations) [2]. The backbone extracts spatiotemporal features from heterogeneous inputs, applies feature fusion, and performs classification at scale. However, supporting both task-specific classification and environment-specific feature extraction inevitably increases model complexity beyond the resource limits of edge devices [41]. Specifically, we introduce the Cross-Dual Path Attention (CDPA) model as the source in this layer (see Section 3.7) and analyze its computational requirements (see Section 3). Therefore, in our framework, the backbone models are deployed in the cloud, while lightweight models perform domain adaptation and task specialization at the edge, as described in the following section.

1.3. Generalization Layer

A major obstacle in applications related to concurrent pose, gesture, position, and activity estimation using CSI is the susceptibility in the signal space to environmental specific factors. A backbone model that is trained on a specific environment for instance will typically not work well when deployed in a new environment and on a different task. We term these environments (networks) as source domain (network) and as target domain (network). In addition to target tasks, the environment specific factors we focus on include device settings (e.g., number of antennas, physical distances), artifacts (e.g., obstacles and

furniture) as well as receiver-transmitter locations within the room. Any change arising from one of these factors is bound to change the input patterns drastically, making domain adaptation a daunting task. Furthermore, as is the case in practical settings, access to labeled and large amount of new data is not always possible.

To circumvent these challenges, in the generalization layer we introduce a bi-directional learning policy that is able to transfer current learned knowledge from a source to a target environment in one direction. In the opposite direction, environment-independent knowledge is distilled from target networks back in the source model across as many tasks per network as need be. Prior work on generalization has explored different trade-offs between efficiency and performance. In [9], low training times are achieved by increasing the learning rate, though at the cost of reduced accuracy. Federated and ensemble architectures [11,22,42] address this limitation by aggregating locally trained models into a globally converging model, thereby improving network-wide performance. To further enhance efficiency, federated split learning [43] investigates bi-directional knowledge transfer between partial model segments rather than entire models. More recently, layered architectures [44] have been proposed, where functional layers are built on top of elementary modules to improve flexibility and generalization across diverse applications.

We adopt a transfer and mutual distillation paradigm that enables weight sharing within architectures and supports multiple tasks across diverse models. Similar to [45], our framework extracts environment-independent features, and as in [43], it facilitates effective cross-network knowledge sharing. Unlike prior work, we focus on model and task diversity by leveraging lightweight architectures tailored to edge resource constraints [46]. For a given task in one network, different light-weight models are trained in a distributed fashion (i.e., edge processing). The knowledge of such models is shared via mutual distillation for different architectures and via transfer learning for similar architectures. In order to enable effective learning across tasks, a larger model is introduced with the capacity to transfer knowledge across networks. Given its computational demands, this backbone is maintained in the cloud as mentioned above, while edge models benefit from its distilled guidance for efficient local inference. Our layered framework aligned with the cloud–edge infrastructure is illustrated in Figure 2 and described in detail in the following section.

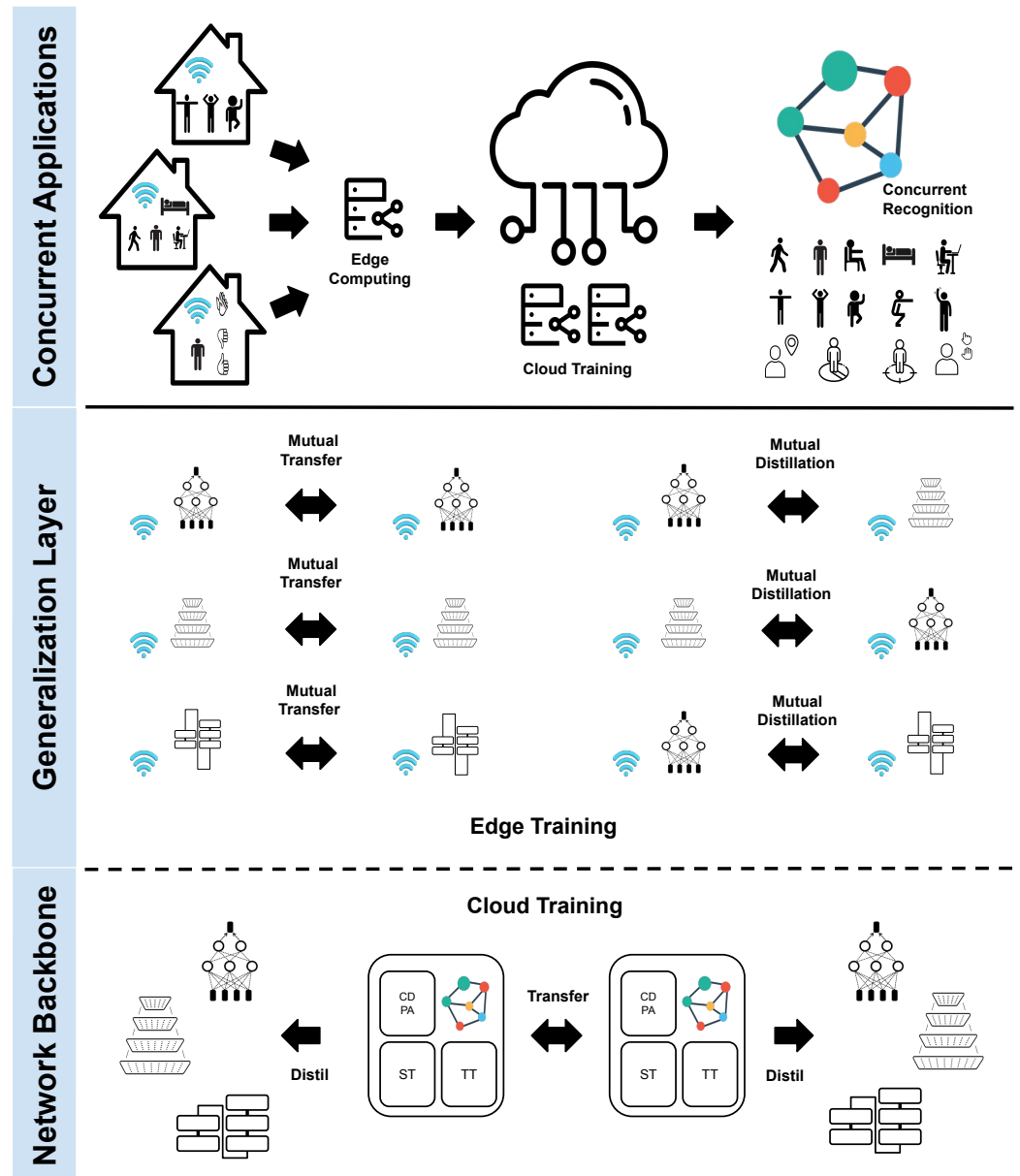


Figure 2. Illustration of multi-environment edge computing and multi-task server processing for concurrent recognition (**top**). Edge devices are equipped with light-weight, diversified models that communicate with a source model, i.e., the CDPA architecture (**middle**), through mutual distillation and transfer learning. In one round of communication, knowledge is shared among edge devices as well as CDPA models in the cloud through bidirectional transfer and distillation according to Algorithm 1 (**bottom**).

Algorithm 1 Extended Mutual Distillation and Transfer Learning

Require: Local model parameters $\{\theta_t^a\}$ for tasks $t \in \{1, \dots, T\}$ and model architectures $a \in \{1, \dots, A\}$, global parameters $\{\theta_a\}$, learning rate η , distillation coefficient α
Ensure: Updated global parameters $\{\theta_a\}$ and local parameters $\{\theta_t^a\}$

- 1: **for** $t = 1$ to T **do** ▷ Local Mutual Distillation
- 2: **Initialize:** set of model indexes $\mathcal{A}_t$ for task t
- 3: **for** $a \in \mathcal{A}_t$ (set of architectures for task t) **do**
- 4: $\mathcal{L}_{KL} \leftarrow 0$
- 5: **for** $a' \in \mathcal{A}_t \setminus \{a\}$ **do**
- 6: $\mathcal{L}_{KL} \leftarrow \mathcal{L}_{KL} - \frac{\eta}{(|\mathcal{A}_t| - 1)} \mathcal{D}_{KL}(\bar{p}_{a'} \parallel p_a)$

Algorithm 1 *Cont.*

```

7:   end for
8:    $\theta_t^a \leftarrow \theta_t^a + (1 - \alpha) \nabla_{\theta_t^a} \mathcal{L}_{\text{CE}}^t + \alpha \nabla_{\theta_t^a} \mathcal{L}_{\text{KL}}$ 
9:   end for
10: end for

11: for  $a = 1$  to  $A$  do ▷ Global Aggregation
12:   Initialize: set of model indexes  $\mathcal{T}_a$  with architecture  $a$ 
13:    $\Delta \bar{\theta}_a \leftarrow \vec{0}$ 
14:   for  $t \in \mathcal{T}_a$  (set of tasks for architecture  $a$ ) do
15:     for  $t' \in \mathcal{T}_a \setminus \{t\}$  do
16:        $\Delta \bar{\theta}_a \leftarrow \Delta \bar{\theta}_a + \frac{1}{(|\mathcal{T}_a| - 1)} \nabla_{\theta_{t'}^a} \mathcal{L}_{\text{CE}}^t$ 
17:     end for
18:   end for
19:    $\bar{\theta}_a \leftarrow \bar{\theta}_a + \eta \Delta \bar{\theta}_a$ 
20: end for

21: return  $\{\theta_t^a\}$  and  $\{\bar{\theta}_a\}$ 

```

2. Methodology

In this section, we provide an in depth explanation of the proposed methodology for a practical CSI-based architecture and recognition system with support for multiple tasks and local models. The components of the system are as follows. The input data are first transformed through the backbone layer to extract relevant features. On the server side (i.e., cloud computing) with sufficient resources, a Cross-Dual-Path-Attention (CDPA) architecture is adopted to extract both spatial (to capture subtleties related to certain fine motions e.g., swipe, etc.) and temporal representations (to further capture the inter-subcarrier correlations). In the network layer, where features that contribute to recognition task (e.g., activity, gesture, pose, direction, etc) minus environment specific effects are to be shared, multiple light weight architectures are deployed (i.e., edge computing). The backbone therefore attempts to extract environment independent features, whereas, local models in target networks are trained to capture environment specific effects, in a globally aligning fashion. The details of the proposed distributed algorithm are provided in the following sections.

2.1. Model Inputs

Given a pair of receiver and transmitter WiFi antennas, each with N_{rx} and N_{tx} antennas and K sub-carriers, the channel information are collected at a given time as input data. The vector of channel state information is constructed as $\vec{h} = [h_1, \dots, h_K] \in \mathbb{C}^K$. Consider N (discrete) channel measurements, the collected CSI in compact matrix form can be constructed by $\mathbf{H} = [\vec{h}_1, \dots, \vec{h}_N] \in \mathbb{C}^{K \times N}$, where N is the sequence length corresponding to the time index and N the total number of sub-carriers over a multi-antenna, multi-channel communication link (the number of antennas times the number of sub-carriers). The CSI tensor $\mathbf{H}$ consisting of the raw amplitudes, forms the input to our system.

Backbone Model

As the backbone model, we employ two different neural network architectures to extract features from the raw input representations. In the spatial transformation modality (ST), the representations of the raw CSI data are fed into a convolutional operator in order to extract finer features followed by a single layer perceptron (SLP) to produce its final spatial representations $\hat{\mathbf{H}}_S$. The relation between the matrix of original raw and the transformed

matrix of CSI through spatial transformation $\mathcal{S}$, denoted by $\hat{\mathbf{H}}_{\mathcal{S}}$ is given by $\hat{\mathbf{H}}_{\mathcal{S}} = f_{\mathcal{S}}(\mathbf{H})$. Here, $f_{\mathcal{S}}$ is the spatial transformation function, parameterized by $\Theta_{\mathcal{S}}$.

In the temporal transformation modality, we aim to capture the long term temporal dependencies in the input representations. Given the input representation $\mathbf{H}$, the attention map/matrix is given by $\mathcal{T} = \text{softmax}(\mathbf{W}^{\mathcal{T}} \circ \mathbf{H})$, where $\mathbf{W}^{\mathcal{T}}$ are learnable parameters. The new representation is thereby obtained by applying $\mathcal{T}$ to raw input, followed by an addition and normalization operation, and a dense layer (SLP for dimension matching). We also apply a mask in the self-attention map so that $\mathbf{W}^{\mathcal{T}}$ is lower triangular, to ensure forward-in-time causality. The relation between the matrix of original raw CSI representations and the transformed matrix of CSI through attention map $\mathcal{T}$, denoted by $\hat{\mathbf{H}}_{\mathcal{T}}$, becomes: $\hat{\mathbf{H}}_{\mathcal{T}} = f_{\mathcal{T}}(\mathbf{H})$. Here, $f_{\mathcal{T}}$ is the temporal transformation function, parameterized by $\Theta_{\mathcal{T}}$.

Since the spatial and temporal representations from the previous stages are learned in completely separate parameter spaces, naive combining of such representations could make the learning task near impossible as each module tends to update the gradient independently. We utilize the architecture introduced in [12], named Cross-Dual-Path Attention (CDPA), for achieving high accuracy and scalable fusion. The final cross-dual path representation is thereby obtained through an addition and normalization operation followed by a dense layer (i.e., SLP). The relation between the spatial and temporal representations and the resulting cross-modal representation, becomes: $\hat{\mathbf{H}}_{\mathcal{ST}} = f_{\mathcal{ST}}(\hat{\mathbf{H}}_{\mathcal{S}}, \hat{\mathbf{H}}_{\mathcal{T}})$, where $f_{\mathcal{ST}}$ is the cross-modal transformation function, parameterized by Φ .

2.2. Multi-Application Recognition

Thus far in the discussion, we have provided a complete account of all the components needed in order to implement a full architecture for a single application, such as activity recognition, from raw input CSI to predicted labels. Ultimately, we are interested in supporting multiple CSI-based applications concurrently as in a multi-task setting. In this setting, multiple tasks are solved jointly, sharing inductive bias between them [47]. In what follows the terms Application and Task are used interchangeably.

Consider a multi-task learning problem given a large dataset of size N' with input and per-task labels $\{\mathbf{H}_i, y_i^1, \dots, y_i^T\}_{i=1}^{N'}$, where T is the number of tasks and y_i^t is the label of the t th task for input $\mathbf{H}_i$. For the CDPA model $f(\mathbf{H}; \Theta_{\mathcal{S}}, \Theta_{\mathcal{T}}, \Phi)$ and given task-specific cross-entropy loss function $\mathcal{L}_{CE}^t(\cdot, \cdot)$, the empirical risk minimization formulation can be written as follows:

$$\min_{\Theta_{\mathcal{S}}, \Theta_{\mathcal{T}}, \Phi} \sum_{i=1}^{N'} \sum_{t=1}^T c_{i,t} \mathcal{L}_{CE}^t(f(\mathbf{H}_i; \Theta_{\mathcal{S}}, \Theta_{\mathcal{T}}, \Phi), y_i^t), \quad (1)$$

where coefficients $c_{i,t}$ weigh contributions from each input to the t th loss component, with no contribution when $c_{i,t} = 0$. Optimizing the above function results in a fully centralized set of parameters across multiple tasks, and can be performed with GradNorm algorithm [48]. In the multi-network setting of this work, we instead use predictions by models with aggregated parameters per task as soft labels to minimize this loss function through distillation as explained next.

2.3. Edge Models

In this work, we assume that edge devices operate under resource constraints, including memory and processing units. The edge devices are only able to run light-weight computations on locally allocated datasets and communicate with a server for heavy load computations. The local datasets are not to be shared among target models themselves and edge devices may additionally have different architectures. In our multi-task setting, the CDPA model is available on the server only with access to weights and data from

other environments. Deploying a centralized CDPA architecture is deemed infeasible due to limited computational resources of edge devices. In this section we provide details pertaining to the training of local models in individual environments with access to limited training data.

More specifically, a set of models sharing the same network server is considered that with diverse architectures $\mathcal{A} = \{1, \dots, A\}$. Each model $f(\mathbf{H}; \theta_a)$, parameterized by θ_a for architecture $a \in \mathcal{A}$, performs local updates using its subset of training data and task-specific loss function $\mathcal{L}^t(\cdot, \cdot)$. If there are multiple models with the same architecture in a network, their parameters are aggregated using federated averaging as $\bar{\theta}_a$ via a global model. The mutual information between the aggregated network of edge models can be minimized by [49]:

$$\min_{\bar{\theta}_1 \dots \bar{\theta}_A} \sum_{a=1}^A \sum_{a' \neq a} [\mathcal{D}_{\text{KL}}(\bar{p}_a \parallel \bar{p}_{a'}) + \mathcal{D}_{\text{KL}}(\bar{p}_{a'} \parallel \bar{p}_a)], \quad (2)$$

where $\bar{p}_a$ is the output probability distribution of the global model parameterized by aggregate weight $\bar{\theta}_a$, i.e., $p(z) = \exp(z) / \sum_{z' \neq z} \exp(z')$. For global models with distinct architectures a and a' , mutual distillation ensures their knowledge is shared (at the rate of once per communication round).

A local model parameterized by θ_a (of architecture type a) associated with the global model parameterized by $\bar{\theta}_a$, updates its own parameters after receiving the associated global weights. The total loss objective for this model can be derived as the sum of empirical risk over the local dataset and the contribution from above mutual distillation, through:

$$\min_{\theta_a} \sum_{i=1}^N \left[\mathcal{L}_{\text{CE}}^t(f(\mathbf{H}_i, \theta_a), y_i) + \sum_{a' \neq a} \mathcal{D}_{\text{KL}}(\bar{p}_{a'} \parallel p_a) \right], \quad (3)$$

where p_a and $\bar{p}_{a'}$ are the output distributions of the local and global models parameterized θ_a and $\bar{\theta}_{a'}$, for N local data points. In this relation, the first term is the cross-entropy loss $\mathcal{L}_{\text{CE}}$ with respect to own parameters and data (i.e., task and environment specific). The second term is the Kullback–Leibler divergence $\mathcal{D}_{\text{KL}}$ which is calculated as the distance between the output distribution of the local model and other global models (having different architectures and tasks).

2.4. Scalable Learning Algorithm

In what follows we continue the discussion by detailing out an algorithm for training models dispatched in local networks in more practical and dynamic settings. The goal is to train local models on specific tasks in a number of target environments where full dataset labeling is not practical or limited re-training is allowed. From such target domains, the source network (i.e., the CDPA model) is to draw common information and transfer to other target networks its knowledge for feature learning unique to their domain. This process continues as new environment and tasks emerge. In order to support multiple tasks (e.g., activity, pose, etc) per target network and multiple architectures (e.g., MLP, CNN, etc) per task, diverse local and global models are deployed in each target domain. As explained previously, local models are trained on respective tasks and global models aggregate their parameters following multiple communication rounds.

During one round of communication, as shown in Algorithm 1, two sets of updates are performed network wide as follows:

- Step 1: calculate gradients for each local model based on the minimization objective in relation (3). For each task, the objective function is calculated across the global models

assigned to the task (potentially) with diverse architectures plus the task-specific cross-entropy loss value.

- Step 2: obtain global weights by aggregating the local model parameters. For each architecture type, the global weights are obtained by averaging the gradient weights across the local models assigned to different tasks.

It is worth noting that each local model in step 1 above is updated with the gradient of a weighted objective function via hyper-parameter α . Each iteration of the first step is carried out through local data subsets and does not require access to the entire data. The second step requires only the sharing of pre-computed gradients and can therefore be performed efficiently.

At the end of each communication round, the backbone model is updated using the global weights through distillation. A key insight here is that, the source network (i.e., the CDPA model) becomes the statistical centroid of all target distributions. In other words, the above process can be thought of as occurring in the space of distributions (as opposed to parameter space) via:

$$p_0^* = \arg \min_{\Theta_S, \Theta_T, \Phi} \sum_{t=1}^T \mathcal{D}_{KL}(p_0 || p_t), \quad (4)$$

where $*$ signifies the optimal solution in which all target networks achieve maximal performance, for upto T tasks with parameters $[\Theta_S^1, \Theta_T^1, \Phi^1, \dots, \Theta_S^T, \Theta_T^T, \Phi^T]$. Here KL denotes the Kullback–Leibler divergence between two distributions, where p_0 and p_t denote the output probability distributions of the central and task-specific source models (i.e., CDPA models) respectively. We wish to learn a source distribution as close in statistical sense to this optimal solution as possible in order to transfer common features to and from target networks for multi-task, environment specific predictions. The extended mutual distillation and transfer learning described above aims at achieving this goal as shown empirically in the next section.

3. Experimental Results

In this section, we provide experimental results to evaluate the performance of the proposed algorithm using a number of datasets. Each dataset constitutes a target network in which global and local models are trained to perform the same task, i.e., each dataset defines one task. Each dataset is further divided into random splits between the local models. During experimentation the architecture is assigned at random to models. We select up to four architectures ($|\mathcal{A}| = 4$), including a multi-layer perceptron (MLP), a Convolution Neural Network (CNN), an Long Short-Term Memory (LSTM), and a Vision Transformer (ViT). The details, including layer information and dimensions, for each architecture is specified in Table 2. The architecture details are fixed throughout the experiment. While in each target network we experiment with a different number of local models per architecture, a single global model per architecture is used for parameter aggregation (i.e., four global models per target network).

We train a CDPA model for each target network to distill the global and local models, with the following components. We employ a ResNet18 stem (the ST block) and a Vision Transformer stem (the TT block) with $\text{emb_dim} = 120$, $\text{depth} = 1$, and $\text{num_heads} = 1$. The model is trained asynchronously for 100 internal epochs on an Apple M2 computer to minimize the loss function in relation (1).

Table 2. Description of the four light-weight architectures used for model training and mutual distillation. Average inference and train times (in seconds per a batch of size 100) as well as the number of add and multiply operations (in mega flops) across all tasks are included.

Model	Inference	Train	FLOPs	Hidden Dimension	Layer Specifications
MLP	0.004	0.82	8.61	123	Conv2d (3×3) $\rightarrow$ BatchNorm2d $\rightarrow$ ReLU $\rightarrow$ MaxPool2d (2×2) $\rightarrow$ Linear (dynamic) $\rightarrow$ ReLU $\rightarrow$ Linear (dynamic) $\rightarrow$ AdaptiveAvgPool2d (8×8)
ViT	0.017	1.84	381	64	Conv2d (3×3) $\rightarrow$ BatchNorm2d $\rightarrow$ ReLU $\rightarrow$ MaxPool2d (2×2) $\rightarrow$ MultiheadAttention (4) $\rightarrow$ AdaptiveAvgPool2d (8×8)
CNN	0.002	0.34	16.6	64	Conv2d (3×3) $\rightarrow$ BatchNorm2d $\rightarrow$ ReLU $\rightarrow$ MaxPool2d (2×2) $\rightarrow$ Conv2d (3×3) $\rightarrow$ BatchNorm2d $\rightarrow$ ReLU $\rightarrow$ Conv2d (3×3) $\rightarrow$ BatchNorm2d $\rightarrow$ AdaptiveAvgPool2d (8×8)
LSTM	0.011	0.58	30.5	128	Conv2d (3×3) $\rightarrow$ BatchNorm2d $\rightarrow$ ReLU $\rightarrow$ MaxPool2d (2×2) $\rightarrow$ LSTM (64,128) $\rightarrow$ AdaptiveAvgPool2d (8×8)

3.1. Datasets

In this paper, we utilize publicly available data for training and evaluation consisting of CSI for different tasks. In particular, one dataset per task is considered as summarized in Table 3. The datasets comprising varying equipment settings, subjects, and environment configurations, for each of $T = 10$ tasks are:

Activity Recognition (AR)—datasets used for the task of human activity recognition are:

- **StanWiFi** [50] with seven activities including “lie down”, “fall”, “walk”, “run”, “sit down”, “stand up”, and “pick up”. These activities were performed twenty times by 6 subjects. Each data frame is 500 (the number of samples) by 90 (the number of subcarriers), by 90 (the number of timestamps).
- **Apartment** [10] with 4 activities including “pickup”, “sitdown”, “standup”, and “walk” performed by a single subject, and a null class where no subject is present. Each data frame is collected over 156 sub-carriers with the dimensions as above.
- **E-EYE** [51] with five different activities, “falling”, “standing”, “walking”, “sitting down” and “standing up” from a chair, and “picking a pen” from the ground, each repeated 20 times. In total, 3000 samples were collected corresponding to 30 subjects, 5 experiments per subject performed for 20 times each.
- **MultiE** [9], including 6 different activities (“wiping”, “walking”, “moving”, “rotating”, “sitting”, and “standing up”), performed by 6 different subjects across 100 different environment settings (e.g., furniture and location variations).

Gesture Recognition (GR)—the datasets used for the task of gesture recognition are:

- **NTU-Fi** [52], comprising classes “running”, “walking”, “falling”, “boxing”, “circling arms”, and “cleaning” the floor—performed by 20 volunteers repeated 20 times. Each data frame spans a duration of 1 s of size 3 by 114 by 500.
- **Widar 3.0** [53]—CSI dataset designed for human gesture recognition, participants performed 22 distinct gestures, including drawing numbers from 0 to 9 in the horizontal plane. The dataset includes an extensive collection of 43,000 samples, providing a robust foundation for studying gesture recognition in diverse scenarios. The CSI frames are pre-processed according to body velocity profile.

- **SignFi** [54]—designed for CSI-based sign language gesture recognition focusing on 150 different gestures by five participants in two environments. Each data frame has dimensions of $3 \times 30 \times 200$.

Pose Recognition (PR) the datasets used for the task of pose and orientation recognition are:

- **mmWPose** [35]—the dataset for CSI-based pose estimation with three participants (two males and one female, with varying body shapes and heights) performing a set of eight poses: “Arms up”, “Left hand up”, “Right lean”, “Right hand up”, “Left lean”, “Empty”, and “Arms wide”. Each pose was held for 15 s, and each participant performed 20 rounds of the pose set, yielding approximately five minutes of data per pose.
- **DirWiFi** (<https://gitlab.com/yuxiqin/direction-independent> (accessed on 12 December, 2024))—the dataset for CSI-based direction estimation with five distinct hand gestures: “drawing a circle”, “crossing hands”, “clapping”, “raising hands”, and “lowering hands”, corresponding to actions like starting, stopping, switching context, increasing, and decreasing. Gestures were performed in 24 directions sampled at 15° intervals, with a natural deviation of approximately $\pm 5^\circ$ in gesture direction. Three volunteers performed the five gestures 40 times in each of the 24 directions.

Fingerprinting/Localization (LFP)—the dataset used for the task of localization based on CSI finger printing is:

- **FP-Loc** [55]—the datasets for CSI-based localization collected in two distinct indoor environments: a laboratory and a meeting room, to represent both Non-Line-of-Sight (NLOS) and Line-of-Sight (LOS) scenarios, consisting of 317 locations spaced 50 cm apart and of 176 locations spaced 60 cm apart, respectively.

Table 3. Dataset Descriptions in terms of the number of antennas, subcarriers, environments, subjects, classes and tasks—AR: Activity Recognition, GR: Gesture Recognition, PR: Pose Recognition, LFP: Location Finger-Printing.

Dataset	Antennas	Subcarriers	Environments	Subjects	Classes	Task
StanWiFi	3	30	1	6	7	AR
Apartment	3	52	2	1	5	AR, LFP
E-Eye	3	30	3	30	12	AR, LFP
MultiE	2	30	6	11	6	AR
NTU-Fi	3	114	1	20	6	GR
Widar 3.0	3	165	1	2	22	GR
SignFi	3	50	2	5	150	GR
mmWPose	30	50	1	3	8	PR
DirWiFi	3×3	30	1	3	24	PR, GR
FP-Loc	3	30	2	1	21×23	LFP

3.2. Network-Wide Performance

In the first set of experiments, a group of local models with randomly selected architectures are assigned to each task. Each local model minimizes the loss function in relation (3) on a task-specific subset and one global model is assigned to aggregate model parameters per task (i.e., AR, GR, PE, or LFP). The local models and the global weights are updated following the steps shown in Algorithm 1. We deploy a total of sixteen local models to train on a given task along with a global model per architecture for weight averaging. Each local model is assigned and initialized at random with a fixed architecture from one of four architecture types specified in Table 2. During one round of the algorithm, the local models are updated based on a combination of the cross-entropy loss and mutual distillation loss with respect to task-specific subset of data and the corresponding global

weight. Subsequently, weight aggregation is performed (per architecture type) across tasks to update the parameters for the global models. After each round of the algorithm, the source models (i.e., the CDPA models) are distilled using the global soft labels according to relation (1) with equal weight contributions from each task. This process is visualized in Figure 3, where in this case the network parameters are set as $A = 4$ and $T = 10$.

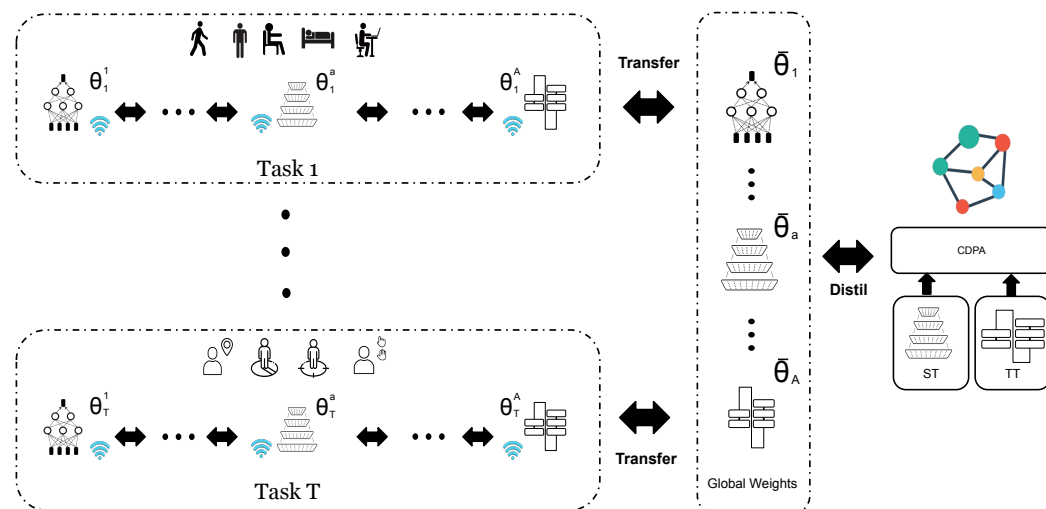


Figure 3. Extended Mutual Distillation and Transfer Learning (EMDTL) framework from Algorithm 1, illustrating local mutual distillation to update task-specific local model parameters θ_t^a for tasks t and architectures a via knowledge sharing, and global weight aggregation to update architecture-specific global parameters $\bar{\theta}_a$ for enhanced generalization across diverse environments.

The main experiment results in terms of source, local and global model accuracies are shown in Figure 4 over 20 rounds of communication using Algorithm 1. We evaluate the performance of the proposed approach (termed EMDTL for Extended Mutual Distillation and Transfer Learning) in multiple settings, including a Fully Centralized setting (shown in dashed line with access to all datasets and using fully supervised objective function), with Mutual Distillation component only (MD), and with Transfer Learning (TL) using global weights only (i.e., no distillation). In each setting, a communication round is defined as one complete pass through all training data, tasks and model architectures, with training data randomly partitioned across models.

It can be observed that through EMDTL the source and local models converge and achieve accuracies approaching the fully centralized scenario. Eventually, the local models tend to outperform the source models due to fine-tuning capability on local data. Furthermore, it can be observed that EMDTL achieves the highest convergence rate compared with the settings in which either MD or TL is not applied. With mutual distillation only (MD), the convergence and accuracies significantly drop with performance gaps of 40.9% and 28.8% between respective source and local models. With global weight sharing only (TL), the gaps are reduced (by 16.7% and 12.3%), however not fully recovered due to the slow convergence of federated averaging on heterogeneous datasets [56,57]. The extraction and sharing of knowledge via bidirectional distillation particularly is found to alleviate conflicting updates [58], with mutual distillation accounting for 24.2% and 16.5% percent performance increase within 20 rounds of EMDTL communications.

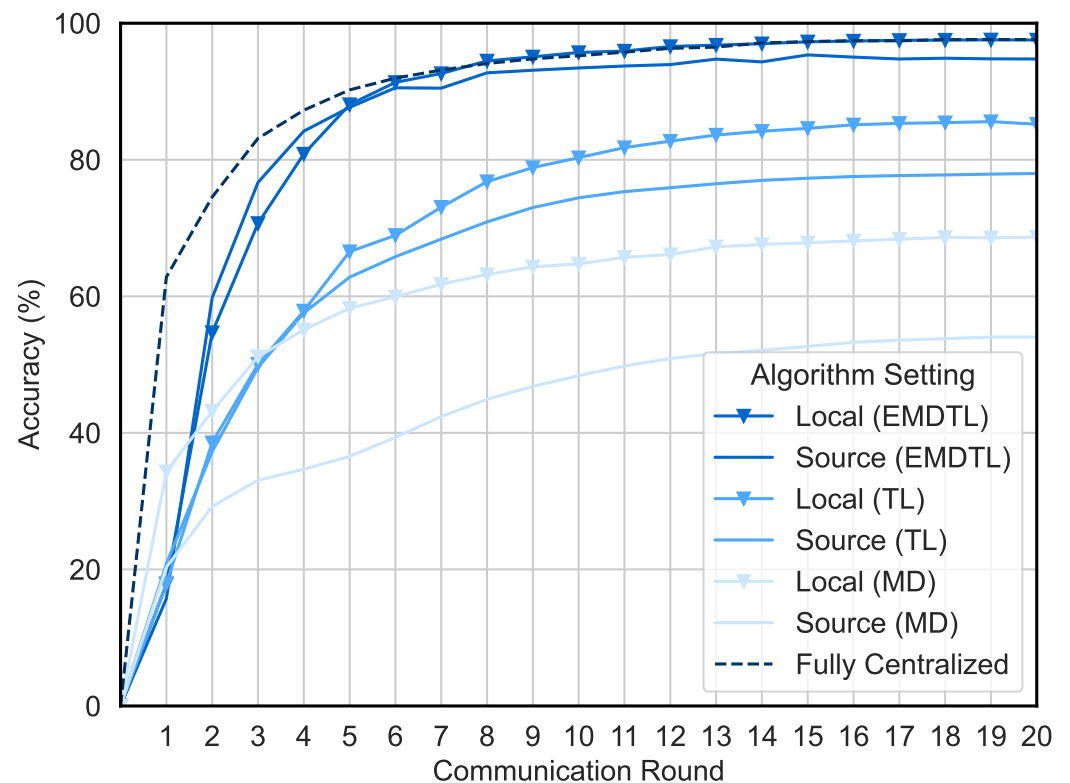


Figure 4. Performance achievement over 20 communication rounds and $T = 10$ applications, in terms of the average accuracies of local (dotted lines) and source (solid lines) models compared to a fully centralized scenario (dashed line), with mutual distillation (MD) only, global weight transfer (TL) only, and the proposed extended mutual distillation and transfer learning (EMDTL).

3.3. Computational Capacity

To evaluate the computational efficiency of the CDPA model relative to lightweight baselines, we measured inference time, training time, and the number of floating-point operations (FLOPs). The CDPA model requires 937 million FLOPs, with a mean inference time of 0.043 s and a training time of 2.86 s per batch of 100 across tasks. These values are significantly higher than the operational capacity of edge devices (see Table 2), which can more readily support simpler models such as LSTM (0.011 s inference, 0.58 s training at 30 M FLOPs), MLP (0.004 s inference, 0.82 s training at 8.61 M FLOPs), CNN (0.002 s inference, 0.34 s training at 16.6 M FLOPs), and ViT (0.017 s inference, 1.84 s training at 381 M FLOPs). These comparisons underscore the importance of the proposed layered framework in balancing efficiency and robustness for energy- and resource-constrained applications. In particular, by leveraging distributed training and knowledge transfer, lightweight edge models can perform local computations while benefiting from the CDPA model trained on a cloud server.

3.4. Effect of Feature Fusion

In order to better understand the effect of cross-dual path attention, we conduct a separate experiment in which the CDPA model is evaluated with different fusion strategies under central training, namely addition and concatenation. We also investigate the performance of the CDPA model when the temporal module (TT) or the spatial module (ST) are activated for inference, where a single layer perceptron (SLP) is applied in the classifier head. Figure 5 depicts the results in terms of test accuracies averaged across all tasks compared to the average state-of-the-art performance (horizontal dotted line). It can be seen that the performance using only single-path TT or ST is inferior to the dual path counter parts, implying the need for both spatial and temporal feature extraction to capture

rich CSI patterns. Fusion with the addition module also under performs compared with the state-of-the-art performance at 93% accuracy. While fusion with concatenation surpasses this metric, it is outperformed by the cross-dual-path-attention mechanism, which achieves the highest accuracy due to better feature integration.

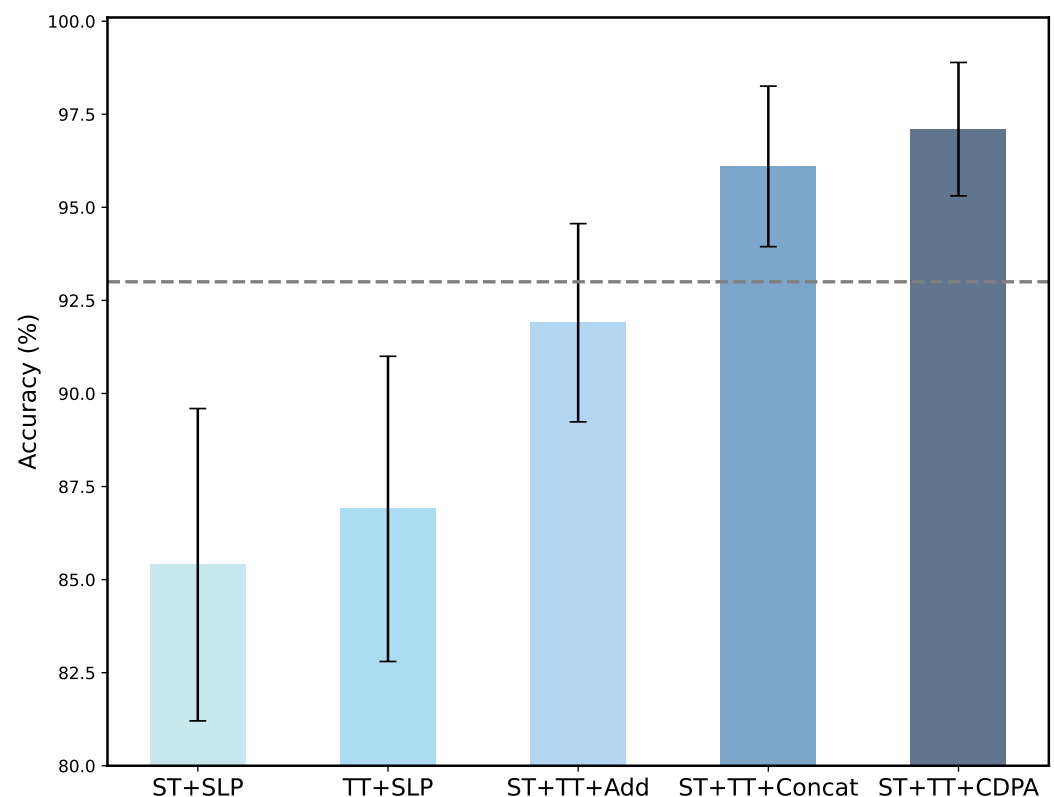


Figure 5. Improvements in concurrent accuracy (mean and standard deviation) through different model configurations, including Spatial Transformer (ST) or Temporal Transformation (TT) with a Single Layer Perceptron (SLP) classifier, or with different fusion strategies including addition (add), concatenation (Concat), or Cross-Dual Path Attention (CDPA). The average state-of-the-art accuracy across all tasks is shown by the dashed line.

3.5. Performance by Edge Models

In terms of local models, we further evaluate and report on the performance for an instance of each architecture in Table 4. Specifically, we report top accuracy over the different task-specific networks with the same four algorithm settings as above, namely centralized training, as well as, with mutual distillation (MD) only added, with global weight transfer (TL) added, and with both added in the extended mutual distillation and transfer learning (EMDTL) setting. In the case of centralized training where individual models are trained on all training dataset from a respective task and evaluated on a held out test set, we observe a large variation in performance across the board. This alludes to the architecture dependent nature of tasks, where on certain datasets one model architecture performs better than others and vice versa. For instance, on SignFi dataset it can be observed that the ViT model performs best; whereas on MultiE dataset the CNN model achieves the best performance (when centrally trained). Across all datasets, the minimum and maximum performances by individual models are 45.9% and 100% for MLP architecture, 80.2% and 100% for CNN architecture, 50.5% and 94.3% for LSTM architecture, and 56.0% and 99.5% for ViT architecture, with centralized training.

Table 4. Performance of individual local architectures over different task-specific networks. For each architecture (MLP, CNN, LSTM, and ViT), the top accuracy (%) is shown when an instance model is trained centrally, with mutual distillation only (MD), with global weight aggregation (TL), or with extended mutual distillation and weight aggregation (EMDTL).

Model	StanW	Apt	EEye	MultiE	NTU-Fi	Widar	SignFi	WPose	DirW
MLP	80.7	98.6	74.7	45.9	96.2	95.6	91.7	100	75
+ MD	98.5	99.2	76	67.1	98.1	95.7	99.9	94	83.3
+ TL	94.9	98.8	84.6	70.6	95.1	98.1	100	99.7	87.5
+ EMDTL	90.9	99.6	90.3	78.4	98.4	98.3	99.8	99.3	96.8
CNN	85.6	100	83.3	89	91.7	92.9	88.7	87.5	81.6
+ MD	91.3	100	85.2	89.7	97.7	93.8	92.1	96.1	89.5
+ TL	96.3	99.6	93.3	81.3	99.6	97.5	99.8	98.8	97.9
+ EMDTL	98.5	100	95.2	91.1	99.2	98.2	99.8	99.8	97.4
LSTM	84.1	73.7	77.5	50.5	82.4	84	94.3	68.6	51.9
+ MD	88.4	87.5	78.6	69.9	92.4	89.3	98.1	89.6	89.6
+ TL	86.9	90.9	82.8	64.7	96.5	84	100	96	94.8
+ EMDTL	94.5	95	90.6	66.8	99.6	96.5	100	97.2	97.9
ViT	70.2	99.5	77.8	48.9	89.3	87.4	99.9	56.0	82.3
+ MD	77.1	97.7	88.9	75.8	90.1	82.6	100	95.2	92.7
+ TL	80.4	100	84.7	75	91.6	97.3	100	97.8	90.6
+ EMDTL	94.2	100	91.8	88.3	92.8	96.3	100	98.9	93.7

With the addition of mutual distillation and transfer learning via global weight sharing, we observe that individual model performances consistently increase in accuracy percentage. This observation is further depicted in Figure 6 in terms of average performance across all test datasets by an individual lightweight model. It can be seen that for each architecture, the model performance improves with the addition of mutual distillation and global weight transfer. Across all tasks and models, the EMDTL results improve upon the basic model architecture performance due to contribution of each component. The minimum and maximum performances by individual models in this case improve to 78.4% and 99.9% for MLP architecture, 91.1% and 100% for CNN architecture, 66.8% and 100% for LSTM architecture, and 88.3% and 100% for ViT architecture. Therefore the individual model architectures trained on edge devices can benefit from our proposed approach as opposed to the scenario in which they are centrally trained and deployed afterwards.

3.6. Effect of Architecture Diversity

To further understand the effect of architecture diversity, a third set of experiments is conducted in which the number of individual models per architecture is varied over task-specific networks. In particular, in this set of experiments we increase the number of models trained between five and forty for any given task. Each local model is assigned a random architecture, hence increasing model diversity per task-specific network, and a random split of training data (equally split between models). Global models are deployed for weight sharing and their performance is measured following the EMDTL algorithm over a held-out test set from each task. The performance is compared to the scenario where a single architecture (i.e., no model diversity) is present, as shown in Figure 7. It can be observed that there is a growing gap between the two settings where the global performance is increased at a higher rate with the increasing number of clients. This effect can be attributed to the diversification of local models (by randomly selecting one of MLP, LSTM, ViT, or CNN architectures). Therefore, the network performance can be improved through global weight sharing with increasing number of diverse models.

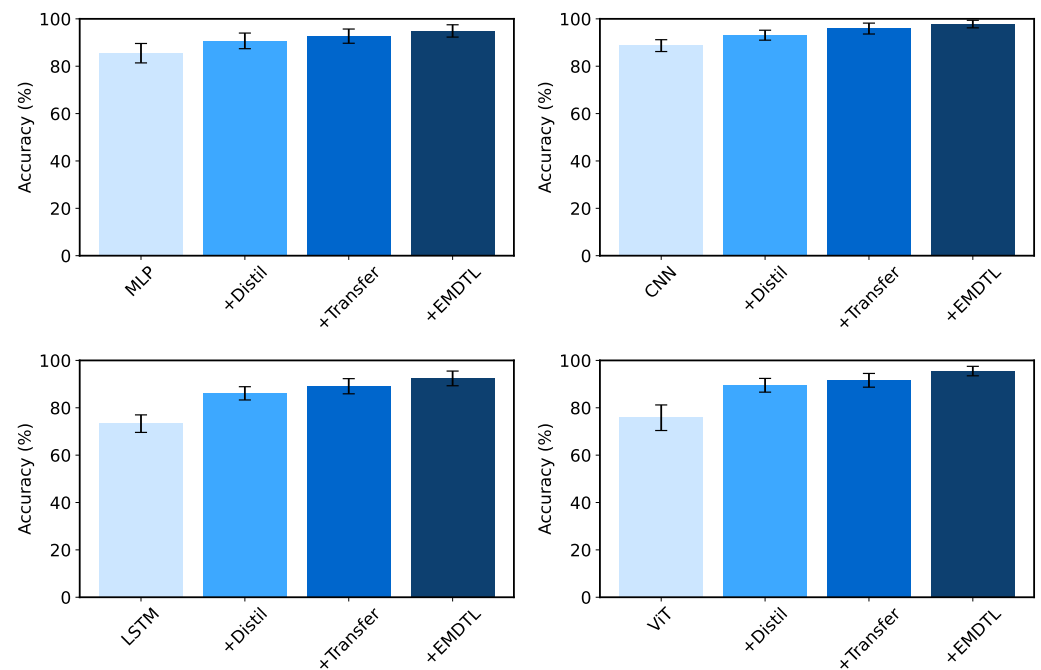


Figure 6. Average performance improvements through the addition of mutual distillation, global weight transfer, and extended mutual distillation and transfer learning (EMDTL), by local models with diverse architectures, namely, Multilayer Perceptron (MLP) (**top left**), Convolution Neural Network (CNN) (**top right**), Long-Short-Term-Memory (LSTM) (**bottom left**), and Visual Transformer (**bottom right**).

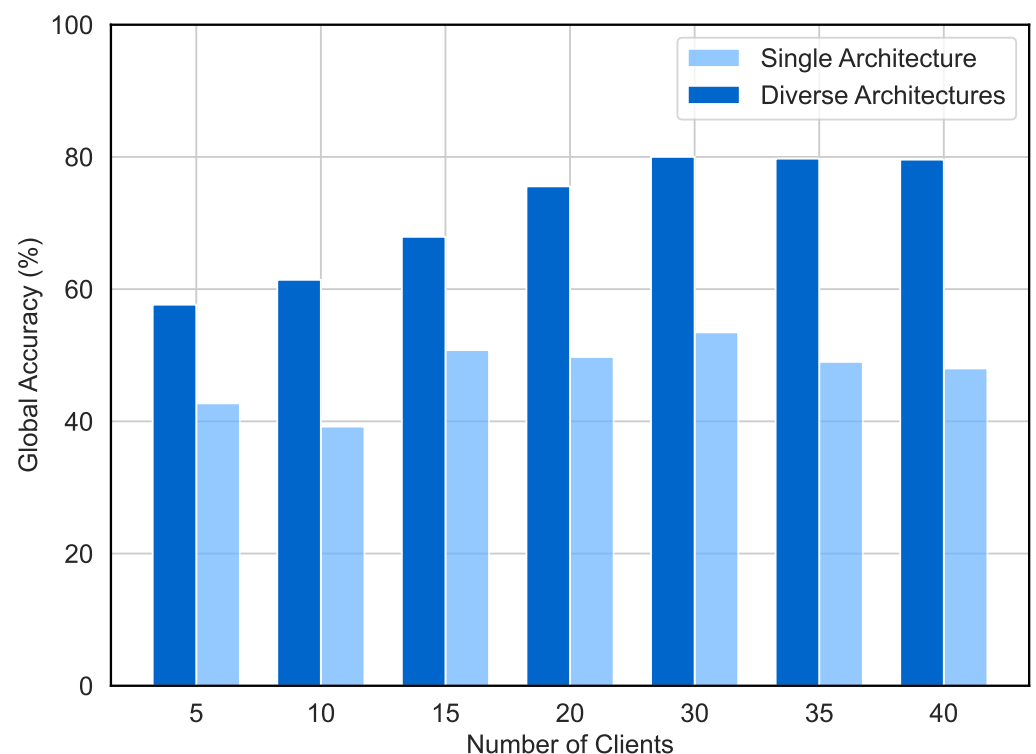


Figure 7. Performance gap between global model accuracies as a function of the number of local models with single architecture type versus diverse architecture types.

3.7. Concurrent Performance by CDPA

Lastly, we report and compare the accuracies of source models (i.e., the CDPA architecture) with state-of-the-art methods (see Table 1) on each dataset. Along with the accuracies

for predictions, we also report precision, recall, and f1-scores where available (see Table 5). For the task of human activity recognition, the CDPA achieves on average 95.2% accuracy, 0.94 precision, 0.95 recall, and 0.94 f1 score on StanWiFi, Apt, E-Eye, and MultiE datasets, on par with the single-task methods. The method of PA-CSI has the best results on both StanWiFi (99.9%) and E-Eye (98.8%) due to specialized amplitude-phase fusion and GRN-based attention. The proposed CDPA model on the hand leverages distill-and-transfer learning and on MultiE (with 6 different environments) outperforms Rf-net and excels on Apt dataset (with 3 different rooms) driven by feature reuse.

Table 5. Comparison of source models with the Cross Dual-Path Architecture (CDPA) against State-of-the-Art (SOTA) reported in the literature in terms of Accuracy, Precision, Recall, and F1 Score. The best and second best results in each case are bolded and underscored, respectively.

Dataset	Model	Accuracy	Precision	Recall	F1 Score
StanW	CNN-GRU	99.3	0.99	0.99	-
	STC-NLSTMNet	99.8	0.99	0.99	-
	AAE+RF	<u>99.8</u>	<u>0.99</u>	<u>0.99</u>	-
	PA-CSI	99.9	0.99	0.99	0.99
	CDPA (ours)	97.0	0.94	0.96	<u>0.95</u>
Apt	LA-DCML	<u>98</u>	<u>0.98</u>	<u>0.98</u>	<u>0.98</u>
	CDPA (ours)	100	1.00	1.00	1.00
EEye	AAE+RF	93.3	0.93	0.93	0.93
	STC-NLSTMNet	<u>94.7</u>	<u>0.94</u>	<u>0.95</u>	<u>0.95</u>
	PA-CSI	98.8	0.98	0.98	0.99
	CDPA (ours)	94.0	0.93	0.93	0.93
MultiE	RF-net	<u>71.0</u>	-	-	-
	CDPA (ours)	90.0	0.90	0.90	0.90
NTU-Fi	BiLSTM	92.1	0.93	0.92	0.92
	MSF-Net	99.5	0.52	0.51	0.76
	CNN-GRU	99.2	<u>0.99</u>	<u>0.99</u>	<u>0.99</u>
	RGANet	<u>99.2</u>	0.99	0.99	0.99
	CDPA (ours)	96.0	0.96	0.96	0.96
Widar 3.0	MSF-Net	82.1	0.50	0.46	0.47
	WiTransformer	86.2	-	-	0.87
	DiffAR	92.1	0.92	-	0.92
	ImgFi	99.8	0.99	0.99	0.99
	CDPA (ours)	<u>96.0</u>	<u>0.96</u>	<u>0.96</u>	<u>0.96</u>
SignFi	MSF-Net	95.5	0.61	0.59	0.60
	AB-LSTM	94.5	0.94	<u>0.95</u>	0.94
	DiffAR	<u>98.2</u>	<u>0.99</u>	-	<u>0.98</u>
	CDPA (ours)	100	1.00	1.00	1.00
WPose	MM-Pose	93.6	-	-	-
	GAN-mmWave	<u>97.7</u>	-	-	-
	CDPA (ours)	99.0	0.98	0.98	0.98
DirWi	WiNDR	<u>92.4</u>	-	-	-
	CDPA (ours)	100	1.00	1.00	1.00

For the task of gesture recognition, the CDPA achieves on average 97.3% accuracy, 0.97 precision, 0.97 recall, and 0.97 f1 score on NTU-Fi, Widar 3.0, and SignFi datasets, on

par with the single-task methods. MSF-Net utilizes dual-stream fusion through a vision transformer applied to Fourier (STFT) and Wavelet (DWT) transformations of the input tailored for gesture recognition. It achieves the highest accuracy on NTU-Fi with degraded performance on Widar 3.0. On the latter dataset, ImgFi performs Gramian Angular Sum and Difference transformations among other techniques and is able to achieve improved performance over the CDPA model on this single task. The CDPA model excels on the SignFi dataset (with 150 classes), achieving the highest performance due its distill-and-transfer strategy, leveraging local models to enhance multi-class gesture recognition.

Across pose and direction classification, improvements by 4 and 8 percent are achieved by the proposed architecture on mmW Pose and DirWiFi, respectively. On CSI-based fingerprinting dataset FP-Loc, the CDPA model excels at outputting discrete coordinates of subject locations achieving a near-perfect accuracy at 99.9%, though results are omitted from Table 5 due to the lack of state-of-the-art methods.

Overall, the proposed architecture achieves 97% accuracy and 0.97 f1-score on average through concurrent learning. As evident from these results, diversification via local models discussed previously enables efficient and robust training across the different tasks. The dual-path design with cross attention enables robust feature fusion, while the distill-and-transfer strategy leverages the mutual information learned among local models in order to achieve scalability and generalizability across tasks. The CDPA-based architecture as such is able to integrate local knowledge and on individual tasks perform comparable with or in some cases better than the state-of-the-art. Deployed on cloud servers, the backbone supports lightweight edge models for efficient inference which in this study are deployed uniformly across tasks. This fixed strategy may limit network-wide performance and dynamic resource allocation as explored in [59,60], is left for future research direction.

4. Conclusions

The omnipresence of WiFi signals presents a great opportunity for a multitude of applications indoors based on the channel state information. In this work, we present a novel algorithm which leverages the mutual information among local models with diverse architectures, for concurrent learning. The architecture supports a wide range of tasks including human activity, gesture, pose, orientation and location estimation in smart environments using passive devices. A layered architecture is proposed in which task and environment specific features can be shared between network devices through a bi-directional policy. Model diversification is found to improve the global performance while concurrent learning is shown to enhance generalization. Extensive experimentation shows that lightweight models usable in edge computing can achieve performance comparable with state-of-the-art single-task methods.

Funding: This research received no external funding.

Data Availability Statement: All datasets are publicly available through linked and provided references.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following notations are used in this manuscript:

Symbol	Description
N_{rx}	Number of Receiver Antennas
N_{tx}	Number of Transmitter Antennas
K	Number of Sub-carriers

$\vec{h}$	Vector of Channel State Information
N	Sequence Length
$\mathbf{H}$	Input CSI Tensor
$\hat{\mathbf{H}}_S$	Spatially Transformed Tensor
Θ_S	Spatial Transformation Parameters
$\hat{\mathbf{H}}_T$	Temporally Transformed Tensor
Θ_T	Temporal Transformation Parameters
Φ	Cross-Dual-Path-Attention Parameters
T	Total Number of Tasks
A	Total Number of Architectures
θ_a	Local Parameters with Architecture a
$\bar{\theta}_a$	Global Parameters for Architecture a
$\mathcal{A}_t$	Set of Architectures for Task t
$\mathcal{T}_a$	Set of Tasks for Architecture a

References

1. Tulino, A.M.; Lozano, A.; Verdu, S. Impact of antenna correlation on the capacity of multiantenna channels. *IEEE Trans. Inf. Theory* **2005**, *51*, 2491–2509. [\[CrossRef\]](#)
2. Zheng, T.; Chen, Z.; Ding, S.; Luo, J. Enhancing rf sensing with deep learning: A layered approach. *IEEE Commun. Mag.* **2021**, *59*, 70–76. [\[CrossRef\]](#)
3. Nguyen, X.H.; Nguyen, V.D.; Luu, Q.T.; Gian, T.D.; Shin, O.S. Robust WiFi Sensing-Based Human Pose Estimation Using Denoising Autoencoder and CNN with Dynamic Subcarrier Attention. *IEEE Internet Things J.* **2025**, *12*, 17066–17079. [\[CrossRef\]](#)
4. Wang, S.; Song, J.; Lien, J.; Poupyrev, I.; Hilliges, O. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology, Tokyo, Japan, 16–19 October 2016; pp. 851–860.
5. Singh, A.D.; Sandha, S.S.; Garcia, L.; Srivastava, M. RadHAR: Human Activity Recognition from Point Clouds Generated through a Millimeter-Wave Radar. In Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems, Los Cabos, Mexico, 25 October 2019; pp. 51–56.
6. Wang, W.; Liu, A.X.; Shahzad, M.; Ling, K.; Lu, S. Understanding and modeling of wifi signal based human activity recognition. In Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, Paris, France, 7–11 September 2015; pp. 65–76.
7. Ma, J.; Wang, H.; Zhang, D.; Wang, Y.; Wang, Y. A survey on wi-fi based contactless activity recognition. In Proceedings of the 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Toulouse, France, 18–21 July 2016; pp. 1086–1091.
8. Damodaran, N.; Haruni, E.; Kokhkharaova, M.; Schäfer, J. Device free human activity and fall recognition using WiFi channel state information (CSI). *CCF Trans. Pervasive Comput. Interact.* **2020**, *2*, 1–17. [\[CrossRef\]](#)
9. Ding, S.; Chen, Z.; Zheng, T.; Luo, J. RF-net: A unified meta-learning framework for RF-enabled one-shot human activity recognition. In Proceedings of the 18th Conference on Embedded Networked Sensor Systems, Virtual, 16–19 November 2020; pp. 517–530.
10. Mehryar, S. Location-aided activity recognition from channel state information with deep cross-modal learning. In Proceedings of the 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), Nafplio, Greece, 26–29 June 2022; pp. 1–5. [\[CrossRef\]](#)
11. Sozinov, K.; Vlassov, V.; Girdzijauskas, S. Human activity recognition using federated learning. In Proceedings of the 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Melbourne, VIC, Australia, 11–13 December 2018; pp. 1103–1111.
12. Mehryar, S. A Domain Adaptation Framework for Human Activity Monitoring Using Passive Wi-Fi Sensing. In Proceedings of the 2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), Hamburg, Germany, 7–8 October 2023; pp. 263–268. [\[CrossRef\]](#)
13. Wang, W.; Liu, A.X.; Shahzad, M.; Ling, K.; Lu, S. Device-Free Human Activity Recognition Using Commercial WiFi Devices. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 1118–1131. [\[CrossRef\]](#)
14. Cai, Z.; Li, Z.; Chen, Z.; Zhuo, H.; Zheng, L.; Wu, X.; Liu, Y. Device-Free Wireless Sensing for Gesture Recognition Based on Complementary CSI Amplitude and Phase. *Sensors* **2024**, *24*, 3414. [\[CrossRef\]](#)
15. Hao, Z.; Duan, Y.; Dang, X.; Liu, Y.; Zhang, D. Wi-SL: Contactless Fine-Grained Gesture Recognition Uses Channel State Information. *Sensors* **2020**, *20*, 4025. [\[CrossRef\]](#)

16. Shi, Z.; Zhang, J.A.; Xu, R.; Fang, G. Human Activity Recognition Using Deep Learning Networks with Enhanced Channel State Information. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6.
17. Luo, F.; Khan, S.; Jiang, B.; Wu, K. Vision Transformers for Human Activity Recognition Using WiFi Channel State Information. *IEEE Internet Things J.* **2024**, *11*, 28111–28122. [[CrossRef](#)]
18. Xiao, C.; Han, D.; Ma, Y.; Qin, Z. CsiGAN: Robust Channel State Information-Based Activity Recognition with GANs. *IEEE Internet Things J.* **2019**, *6*, 10191–10204. [[CrossRef](#)]
19. Lee, H.; Ahn, C.R.; Choi, N. Fine-grained occupant activity monitoring with Wi-Fi channel state information: Practical implementation of multiple receiver settings. *Adv. Eng. Inform.* **2020**, *46*, 101147. [[CrossRef](#)]
20. Shavit, Y.; Klein, I. Boosting Inertial-Based Human Activity Recognition with Transformers. *IEEE Access* **2021**, *9*, 53540–53547. [[CrossRef](#)]
21. Su, J.; Wen, Z.; Lin, T.; Guan, Y. Learning Disentangled Behaviour Patterns for Wearable-based Human Activity Recognition. In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies; Association for Computing Machinery: New York, NY, USA, 2022; Volume 6, pp. 1–19.
22. Zhu, R.; Xiao, Z.; Li, Y.; Yang, M.; Tan, Y.; Zhou, L.; Lin, S.; Wen, H. Efficient Human Activity Recognition Solving the Confusing Activities Via Deep Ensemble Learning. *IEEE Access* **2019**, *7*, 75490–75499. [[CrossRef](#)]
23. Chen, J.; Xu, X.; Wang, T.; Jeon, G.; Camacho, D. An AIoT Framework with Multimodal Frequency Fusion for WiFi-Based Coarse and Fine Activity Recognition. *IEEE Internet Things J.* **2024**, *11*, 39020–39029. [[CrossRef](#)]
24. Zhou, Y.; Yang, J.; Huang, H.; Xie, L. AdaPose: Toward Cross-Site Device-Free Human Pose Estimation with Commodity WiFi. *IEEE Internet Things J.* **2024**, *11*, 40255–40267. [[CrossRef](#)]
25. Li, T.; Wang, H.; Shao, Y.; Niu, Q. Channel state information-based multi-level fingerprinting for indoor localization with deep learning. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718806719. [[CrossRef](#)]
26. Qian, K.; Wu, C.; Zhang, Y.; Zhang, G.; Yang, Z.; Liu, Y. Widar2.0: Passive Human Tracking with a Single Wi-Fi Link. In Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, Munich, Germany, 10–15 June 2018; pp. 350–361.
27. Chen, P.; Liu, Y.; Li, W.; Wang, J.; Wang, J.; Yang, B.; Feng, G. Semi-Supervised Learning-Enhanced Fingerprint Indoor Positioning by Exploiting an Adapted Mean Teacher Model. *Electronics* **2024**, *13*, 298. [[CrossRef](#)]
28. Bastwesy, M.R.; ElShennawy, N.M.; Saidahmed, M.T.F. Deep learning sign language recognition system based on wi-fi csi. *Int. J. Intell. Syst. Appl.* **2020**, *12*, 33–45. [[CrossRef](#)]
29. Shalaby, E.; ElShennawy, N.; Sarhan, A. Utilizing deep learning models in CSI-based human activity recognition. *Neural Comput. Appl.* **2022**, *34*, 5993–6010. [[CrossRef](#)]
30. Islam, M.S.; Jannat, M.K.A.; Hossain, M.N.; Kim, W.S.; Lee, S.W.; Yang, S.H. STC-NLSTMNet: An improved human activity recognition method using convolutional neural network with NLSTM from WiFi CSI. *Sensors* **2022**, *23*, 356. [[CrossRef](#)]
31. Jannat, M.K.A.; Islam, M.S.; Yang, S.H.; Liu, H. Efficient Wi-Fi-Based Human Activity Recognition Using Adaptive Antenna Elimination. *IEEE Access* **2023**, *11*, 105440–105454. [[CrossRef](#)]
32. Huang, S.; Chen, P.Y.; McCann, J.A. DiffAR: Adaptive Conditional Diffusion Model for Temporal-augmented Human Activity Recognition. In Proceedings of the IJCAI, Macao, China, 19–25 August 2023; pp. 3812–3820.
33. Zhang, C.; Jiao, W. ImgFi: A High Accuracy and Lightweight Human Activity Recognition Framework Using CSI Image. *IEEE Sens. J.* **2023**, *23*, 21966–21977. [[CrossRef](#)]
34. Yang, M.; Zhu, H.; Zhu, R.; Wu, F.; Yin, L.; Yang, Y. WiTransformer: A novel robust gesture recognition sensing model with WiFi. *Sensors* **2023**, *23*, 2612. [[CrossRef](#)]
35. Bhat, N.N.; Sameri, J.; Struye, J.; Vega, M.T.; Berkvens, R.; Famaey, J. Multi-modal pose estimation in XR applications leveraging integrated sensing and communication. In Proceedings of the 1st ACM Workshop on Mobile Immersive Computing, Networking, and Systems, ImmerCom'23, Madrid, Spain, 6 October 2023; Association for Computing Machinery: New York, NY, USA, 2023; pp. 261–267. [[CrossRef](#)]
36. Bhat, N.N.; Berkvens, R.; Famaey, J. CSI4Free: GAN-Augmented mmWave CSI for Improved Pose Classification. In Proceedings of the 2024 IEEE 4th International Symposium on Joint Communications and Sensing (JC&S), Leuven, Belgium, 19–21 March 2024; pp. 1–6. [[CrossRef](#)]
37. Qin, Y.; Sigg, S.; Pan, S.; Li, Z. Direction-agnostic gesture recognition system using commercial WiFi devices. *Comput. Commun.* **2024**, *216*, 34–44. [[CrossRef](#)]
38. Quy, T.D.; Lin, C.Y.; Shih, T.K. Enhanced human activity recognition using wi-fi sensing: Leveraging phase and amplitude with attention mechanisms. *Sensors* **2025**, *25*, 1038. [[CrossRef](#)]
39. Wakili, A.A.; Asaju, B.J.; Jung, W. Evaluating BiLSTM and CNN+ GRU Approaches for Human Activity Recognition Using WiFi CSI Data. *arXiv* **2025**, arXiv:2506.11165.

40. Hu, J.; Ge, F.; Cao, X.; Yang, Z. RGANet: A Human Activity Recognition Model for Extracting Temporal and Spatial Features from WiFi Channel State Information. *Sensors* **2025**, *25*, 918. [\[CrossRef\]](#)
41. Mehryar, S. Concurrent Learning for CSI-Based Applications in Smart Environments. In Proceedings of the 2025 21st International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), Lucca, Italy, 9–11 June 2025; pp. 538–543. [\[CrossRef\]](#)
42. Yang, H.H.; Chen, Z.; Quek, T.Q.S. Server Free Wireless Federated Learning: Architecture, Algorithm, and Analysis. *arXiv* **2022**, arXiv:2204.07609. [\[CrossRef\]](#)
43. Luo, L.; Zhang, X. Federated Split Learning via Mutual Knowledge Distillation. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 2729–2741. [\[CrossRef\]](#)
44. Czarnecki, W.; Jayakumar, S.; Jaderberg, M.; Hasenclever, L.; Teh, Y.W.; Heess, N.; Osindero, S.; Pascanu, R. Mix and Match Agent Curricula for Reinforcement Learning. In Proceedings of the 35th International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 1087–1095.
45. Jiang, W.; Miao, C.; Ma, F.; Yao, S.; Wang, Y.; Yuan, Y.; Xue, H.; Song, C.; Ma, X.; Koutsounikolas, D.; et al. Towards environment independent device free human activity recognition. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, New Delhi, India, 29 October–2 November 2018; pp. 289–304.
46. Imteaj, A.; Thakker, U.; Wang, S.; Li, J.; Amini, M.H. A Survey on Federated Learning for Resource-Constrained IoT Devices. *IEEE Internet Things J.* **2022**, *9*, 1–24. [\[CrossRef\]](#)
47. Caruana, R. Multitask Learning. *Mach. Learn.* **1997**, *28*, 41–75. [\[CrossRef\]](#)
48. Chen, Z.; Badrinarayanan, V.; Lee, C.Y.; Rabinovich, A. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. *arXiv* **2018**. [\[CrossRef\]](#)
49. Zhang, Y.; Xiang, T.; Hospedales, T.M.; Lu, H. Deep mutual learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4320–4328.
50. Yousefi, S.; Narui, H.; Dayal, S.; Ermon, S.; Valaee, S. A Survey on Behavior Recognition Using WiFi Channel State Information. *IEEE Commun. Mag.* **2017**, *55*, 98–104. [\[CrossRef\]](#)
51. Baha’A, A.; Almazari, M.M.; Alazrai, R.; Daoud, M.I. A dataset for Wi-Fi-based human activity recognition in line-of-sight and non-line-of-sight indoor environments. *Data Brief* **2020**, *33*, 106534.
52. Yang, J.; Chen, X.; Zou, H.; Wang, D.; Xu, Q.; Xie, L. EfficientFi: Toward Large-Scale Lightweight WiFi Sensing via CSI Compression. *IEEE Internet Things J.* **2022**, *9*, 13086–13095. [\[CrossRef\]](#)
53. Zhang, Y.; Zheng, Y.; Qian, K.; Zhang, G.; Liu, Y.; Wu, C.; Yang, Z. Widar3.0: Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 8671–8688. [\[CrossRef\]](#) [\[PubMed\]](#)
54. Ma, Y.; Zhou, G.; Wang, S.; Zhao, H.; Jung, W. SignFi: Sign Language Recognition Using WiFi. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*; Association for Computing Machinery: New York, NY, USA, 2018; Volume 2, pp. 1–21. [\[CrossRef\]](#)
55. Zhu, X.; Qiu, T.; Qu, W.; Zhou, X.; Atiquzzaman, M.; Wu, D. BLS-Location: A Wireless Fingerprint Localization Algorithm Based on Broad Learning. *IEEE Trans. Mob. Comput.* **2023**, *22*, 115–128. [\[CrossRef\]](#)
56. Chang, W.T.; Mohamed, M.S.E.; Tandon, R. On the necessity of aligning gradients for wireless federated learning. In Proceedings of the 2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Lucca, Italy, 27–30 September 2021; pp. 226–230.
57. Zhu, H.; Xiao, Y.; Li, Y.; Shi, G.; Krunz, M. SANSee: A Physical-Layer Semantic-Aware Networking Framework for Distributed Wireless Sensing. *IEEE Trans. Mob. Comput.* **2024**, *24*, 1636–1653. [\[CrossRef\]](#)
58. Geng, H.; Deng, D.; Zhang, W.; Ji, P.; Wu, X. Personalized federated learning based on bidirectional knowledge distillation for wifi gesture recognition. *Electronics* **2023**, *12*, 5016. [\[CrossRef\]](#)
59. Benaboura, A.; Bechar, R.; Kadri, W.; Ho, T.D.; Pan, Z.; Sahnoud, S. Latency-Aware and Energy-Efficient Task Offloading in IoT and Cloud Systems with DQN Learning. *Electronics* **2025**, *14*, 3090. [\[CrossRef\]](#)
60. Zhang, S.; Tong, X.; Chi, K.; Gao, W.; Chen, X.; Shi, Z. Stackelberg game-based multi-agent algorithm for resource allocation and task offloading in mec-enabled c-its. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 17940–17951. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.